```
KEYW    CMPIM 'W
        BNE   LIST    BRANCH IF NO W KEY
        JMP   SEACND  WARM CEND ENTRY
EDTCA   CLC           REDIFINE CEND;
        LDAZ  BEGADL  CEND POINTS AT THE
        ADCIM $01     FREE FILE MEMORY
        STAZ  CENDL   LOCATION WITH THE
        LDAZ  BEGADH  LOWEST ADDRESS
        ADCIM $00
        STAZ  CENDH


        BNE   GKEY    BRANCH IF NO K KEY

        BNE   ESCAPE  BRANCH IF NO X KEY

SEMIW   JSR   INITAD  ENTER/DEFINE PARAMETERS
        JMP   BRK     GO TO PME MAIN PROGRAM
INITAD  JSR   RESIN   RESET INPUT BUFFERS
        JSR   MESSA   "BEGAD,ENDAD: " (Y=00!)
        JSR   INPAR   ENTER TWO ADDRESSES
        BMI   INITAD  REPEAT IF NOT PROPERLY DONE
        LDXIM $03
INTA    LDAX  PARAL   PARA=BEGAD=CURAD
        STAX  BEGADL  PARB=ENDAD=CEND
        STAX  CURADL
        DEX           NEXT ADDRESS BYTE
        BPL   INTA    IF ANY
        RTS           N FLAG IS SET
        NOP           NOTE: AFTER THE COLD START
        NOP           ENTRY AND POSSIBLY AFTER
        NOP           THE WARM CEND START ENTRY
        NOP           CEND WILL BE REDEFINED
        NOP
        NOP
ESCAPE  CMPIM 'V
        BNE   INPUT   BRANCH IF NO V KEY
        JMP   LABJUN  BACK TO PM ("JUNIOR")
DECURA

SEACND  JSR   INITAD  ENTER/DEFINE PARAMETERS
SCNDA   LDYIM $00
        LDAIY CURADL  FETCH OPCODE
        CMPIM $77
        BEQ   SCNDB   BRANCH IF EOF FOUND
        JSR   LENACC  GET INSTRUCTION LENGTH
        JSR   NEXT    COMPARE NEXT OPCODE
        BMI   SCNDA   IF ANY
        LDYIM $90
        JSR   MESSA   "NO 77"
        JMP   LABJUN  BACK TO PM ("JUNIOR")
        'N            EXTENSION OF
        'O            TXT LOOK UP TABLE
        '             (TXT: $1750)
        '7
        '7
        $03           EOT
SCNDB                 REDEFINE CEND
```

```
        ORG   $F800       EPROM VERSION


SOURCE LISTING OF A 6502 DISASSEMBLER
WITH A POWERFUL MONITOR

START ADDRESS: $064E  RAM VERSION
START ADDRESS: $FC4E  EPROM VERSION

WRITTEN BY A. NACHTMANN

DATE: 2 SEP. 1981


**********************
DISASSEMBLER FOR 6502
**********************



TEMPS AND POINTERS IN SCRATCHPAD


INSPNT *      $0010  INSTRUCTION POINTER
OPCODE *      $0012  OP CODE BUFFER
RNIB   *      $0013  RIGHT NIBBLE OF THE OP CODE
LNIB   *      $0014  LEFT NIBBLE OF THE OP CODE
EOLNIB *      $0015  0/1 = EVEN/ODD IS THE LEFT NIBBLE
LENGTH *      $0016  LENGTH OF THE INSTRUCTION 1-2-3
ADDR   *      $0017  DESTINATION OF A RELATIVE BRANCH
TEMP   *      $0019
ERRFLG *      $0020  ERROR FLAG
ML     *      $0021  LEFT MNEMONIC
MR     *      $0022  RIGHT MNEMONIC
MEPNT  *      $0023  MESSAGE POINTER
POINT  *      $00FA  DUMP POINTER
PMODE  *      $0025  PAGE MODE FLAG
CNT    *      $0026  LINE COUNTER
HEXFLG *      $0027  HEX/ASCII FLAG



OTHER TEMPS

PARA   *      $1A63  1ST PARAMETER
PARB   *      $1A65  2ND PARAMETER
BRKT   *      $1A7C  KEYBOARD BRK VECTOR


**************************************
DISASSEMBLE 1 INSTRUCTION AND RETURN
**************************************


DISASM LDYIM $00     FETCH OP CODE
```

```
        LDAIY INSPNT
        STA   OPCODE AND SAVE IT
        ANDIM $0F    GET RIGHT NIBBLE
        STA   RNIB   AND SAVE IT
        LDA   OPCODE
        LSRA
        LSRA
        LSRA
        LSRA
        STA   LNIB   GET LEFT NIBBLE
        LSRA         IS LEFT NIBBLE EVEN OR ODD?
        BCC   DISB
        LDAIM $01    SET E/O FLAG

DISA    STA   EOLNIB SAVE THE FLAG
        LDXIM $04
        LDA   RNIB

VALOP   CMPX  NVALA  IS COLUMN VALID?
        BEQ   ERRA   NOT VALID
        DEX
        BPL   VALOP
        LDXIM $19
        LDA   OPCODE

VALOPA  CMPX  NVALB  VALID OP CODE IN ANY COLUMN
        BEQ   ERROR
        DEX
        BPL   VALOPA
        BMI   COLI

DISB    LDAIM $00    RESET E/O FLAG
        BEQ   DISA

ERRA    LDA   OPCODE
        CMPIM $A2    LDXIM IS VALID
        BEQ   LDXIM

ERROR   LDXIM $01      INVALID OPCODES:
        STX   ERRFLG   INSTRUCTION LENGTH
        STX   LENGTH   ONE
        LDXIM $08       ???  IN MCTAB
        JSR   MCFA
        JMP   CENTRM


LDXIM   LDXIM $02    LENGTH = 2
        STX   LENGTH
        LDXIM $61      NOT RELEVANT
        STX   ML       NOT RELEVANT
        LDXIM $30      NOT RELEVANT
        STX   MR       NOT RELEVANT
        JSR   MBFORM   LNIB=A;INDEX=05!
        JMP   IMX      PRINT OPERAND

COLI    LDXIM $01
```

```
        CPX   RNIB   RNIB = 01?
        BNE   ZPZPX

*** INDY-INDX ***

        INX          LENGTH = 2
        STX   LENGTH
        JSR   MAFORM OUTPUT MNEMONIC
        LDA   EOLNIB E=INDX, 0=INDY
        BEQ   INDX

INDY    JSR   PRINT
        =     '(
        =     '$
        =     $03
        LDYIM $01
        LDAIY INSPNT FETCH OPERAND
        JSR   PRBYT
        JSR   PRINT
        =     ')
        =     ',
        =     'Y
        =     $03

CENTRM  CLC          ADJUST INSTRUCTION POINTER
        LDA   INSPNT
        ADC   LENGTH
        STA   INSPNT
        LDA   INSPNT +01
        ADCIM $00
        STA   INSPNT +01
        RTS

*** INDX ***

INDX    JSR   PRINT
        =     '(
        =     '$
        =     $03
        LDYIM $01
        LDAIY INSPNT FETCH OPERAND
        JSR   PRBYT
        JSR   PRINT
        =     ',
        =     'X
        =     ')
        =     $03
        JMP   CENTRM

*** ZPZPX ***

ZPZPX   LDXIM $05
        CPX   RNIB   RNIB = 05?
        BNE   IMABSY
        LDXIM $02
        STX   LENGTH LENGTH = 2
```

```
        JSR   MAFORM
        LDA   EOLNIB E = ZP, 0 = ZPX
        BEQ   ZP

ZPX     JSR   PRINT
        =     '$
        =     $03
        LDYIM $01
        LDAIY INSPNT FETCH OPERAND
        JSR   PRBYT
        JSR   PRINT
        =     ',
        =     'X
        =     $03
        JMP   CENTRM

ZP      JSR   PRINT
        =     '$
        =     $03
        LDYIM $01
        LDAIY INSPNT FETCH OPERAND
        JSR   PRBYT
        JMP   CENTRM

*** IMABSY ***

IMABSY LDXIM $09
        CPX   RNIB   RNIB = 09?
        BNE   ABSABX
        LDA   EOLNIB
        BEQ   IM     E = IM, 0 = ABSY

ABSY    LDXIM $03
        STX   LENGTH LENGTH = 3
        JSR   MAFORM OUTPUT MNEMONIC
ABSYS   JSR   PRINT
        =     '$
        =     $03
        LDYIM $02    2 OPERANDS

AYA     LDAIY INSPNT
        JSR   PRBYT
        DEY
        BNE   AYA
        JSR   PRINT
        =     ',
        =     'Y
        =     $03
        JMP   CENTRM

IM      LDXIM $02
        STX   LENGTH LENGTH = 2
        JSR   MAFORM
IMX     JSR   PRINT
        =     '#
        =     '$
```

```
        =       $03
        LDYIM $01
        LDAIY INSPNT FETCH OPERAND
        JSR     PRBYT
        JMP     CENTRM


*** ABSABX ***

ABSABX LDXIM $0D
        CPX     RNIB    RNIB = 0D?
        BNE     SFTROT
        LDXIM $03       LENGTH = 3
        STX     LENGTH
        JSR     MAFORM OUTPUT MNEMONIC
        LDA     EOLNIB
        BEQ     ABS     E = ABS, 0 = ABSX

ABSX    JSR     PRINT
        =       '$
        =       $03
        LDYIM $02       2 OPERANDS

AXA     LDAIY INSPNT
        JSR     PRBYT
        DEY
        BNE     AXA
        JSR     PRINT
        =       ',
        =       'X
        =       $03
        JMP     CENTRM

ABS     JSR     PRINT
        =       '$
        =       $03
        LDYIM $02       2 OPERANDS

ABA     LDAIY INSPNT
        JSR     PRBYT
        DEY
        BNE     ABA
        JMP     CENTRM

*** SFTROT ***

SFTROT LDXIM $06
        CPX     RNIB    RNIB = 06?
        BNE     SRCONT
        LDXIM $02       LENGTH = 2
        STX     LENGTH
        JSR     MBFORM OUTPUT MNEMONIC
        LDXIM $09
        CPX     LNIB    ZPY INSTRUCTIONS?
        BEQ     ZPY
        LDXIM $0B
```

```
        CPX   LNIB
        BEQ   ZPY
        LDA   EOLNIB
        BEQ   SRA     E = ZP, O = ZPX
        JMP   ZPX

SRA     JMP   ZP

ZPY     JSR   PRINT
        =     '$
        =     $03
        LDYIM $01
        LDAIY INSPNT
        JSR   PRBYT
        JSR   PRINT
        =     ','
        =     'Y
        =     $03
        JMP   CENTRM

*** SRCONT ***

SRCONT  LDXIM $0E
        CPX   RNIB    RNIB = 0E?
        BNE   COLNUL
        LDXIM $03     LENGTH = 3
        STX   LENGTH
        JSR   MBFORM  OUTPUT MNEMONIC
        LDXIM $0B
        CPX   LNIB    ABSY INSTRUCTION
        BEQ   SRCB    ABSY
        LDA   EOLNIB
        BEQ   SRCA    E = ABS, O = ABSX
        JMP   ABSX

SRCA    JMP   ABS

SRCB    JMP   ABSYS

*** COLNUL ***

COLNUL  LDXIM $00
        CPX   RNIB    RNIB = 0
        BNE   COLFOR
        LDA   OPCODE
        CMPIM $00     BRK INSTR.?
        BEQ   COLNUB
        CMPIM $40     RTI INSTR.?
        BEQ   COLNUB
        CMPIM $60     RTS INSTR.?
        BEQ   COLNUB
        CMPIM $20     JSR INSTR.?
        BEQ   COLNUA
        ANDIM $1F
        CMPIM $10     ANY BRANCH INSTR.?
        BEQ   OFFSET  IF YES, COMPUTE OFFSET
```

```
OTHER   LDXIM $02     LENGTH = 2
        STX   LENGTH
        JSR   MCFORM
        JMP   IMX

COLNUA  LDXIM $03
        STX   LENGTH LENGTH = 3
        JSR   MCFORM
        JMP   ABS     ABSOLUTE ADDRESSING

COLNUB  LDXIM $01     LENGTH = 1
        STX   LENGTH
        JSR   MCFORM OUTPUT MNEMONIC
        JMP   CENTRM

OFFSET  LDXIM $02     LENGTH = 2
        STX   LENGTH
        JSR   MCFORM
        LDYIM $01
        LDAIY INSPNT FETCH OFFSET
        BPL   POSOFF PLUS/MINUS BRANCH
        JSR   ADJ     ACCU IS NOT USED: "INSPNT +1"
        EORIM $FF     COMPLEMENT
        STA   TEMP
        SEC
        LDA   ADDR
        SBC   TEMP
        STA   ADDR
        LDA   ADDR    +01 STORE DESTINATION IN ADDR
        SBCIM $00
        STA   ADDR    +01
        JMP   POA

POSOFF  JSR   ADJ
        SEC           "INSPTR +2"
        ADC   ADDR
        STA   ADDR
        LDAIM $00
        ADC   ADDR    +01
        STA   ADDR    +01

POA     JSR   PRINT
        =     '$
        =     $03
        LDXIM $01

POB     LDAX  ADDR
        JSR   PRBYT  OUTPUT DESTINATION ADDR
        DEX
        BPL   POB
        JMP   CENTRM

*** COLFOR ***

COLFOR LDXIM $04
```

```
        CPX   RNIB    RNIB = 04?
        BNE   COLCW
        LDXIM $02     LENGTH = 2
        STX   LENGTH
        JSR   MDFORM OUTPUT MNEMONIC
        LDA   EOLNIB
        BEQ   COLFRA E = ZP, 0 = ZPX
        JMP   ZPX

COLFRA  JMP   ZP

COLCW   LDXIM $0C
        CPX   RNIB    RNIB = 0C?
        BNE   COLOA
        LDXIM $03
        STX   LENGTH LENGTH = 3
        JSR   MDFORM OUTPUT MNEMONIC
        LDA   OPCODE
        CMPIM $6C     JMP (IND) INSTRUCTION
        BEQ   IND
        CMPIM $BC     LDY,X INSTRUCTION?
        BEQ   SPELDY
        JMP   ABS     ELSE ABS ADDRESSING

SPELDY  JMP   ABSX

IND     JSR   PRINT
        =     '(
        =     '$
        =     $03
        LDYIM $02

INDA    LDAIY INSPNT
        JSR   PRBYT  OUTPUT INDIRECT OPERAND
        DEY
        BNE   INDA
        JSR   PRINT
        =     ')
        =     $03
        JMP   CENTRM

*** COLOA ***

COLOA   LDXIM $0A
        CPX   RNIB
        BNE   IMPLD  ONLY IMPLIED ADDRESSING IS NOW POSSIBLE
        LDXIM $01
        STX   LENGTH LENGTH = 1
        LDX   LNIB
        CPXIM $07    CHECK FOR ACCU ADDRESSING
        BCC   ACCU
        JSR   MEFORM
        JMP   CENTRM

ACCU    JSR   MEFORM
        JSR   PRINT
```

```
        =       '
        =       'A
        =       $03
        JMP     CENTRM

IMPLD   LDXIM $01
        STX     LENGTH
        JSR     MFFORM OUTPUT MNEMONIC
        JMP     CENTRM


*********************************
SUBROUTINES OF THE DISASSEMBLER
*********************************


*** XSPACE ***

XSPACE  JSR     PRSP    OUTPUT X SPACES
        DEX
        BNE     XSPACE
        RTS

*** PRBYTS ***

PRBYTS  JSR     PRBYT   OUTPUT THE BYTE IN A
        JMP     PRSP    AND ADD A SPACE

*** OBJECT ***

OBJECT  JSR     CRLF    OUTPUT OBJECT CODE
        LDA     INSPNT  +01 OUTPUT ADDR. OF THE
        JSR     PRBYT   CURRENT OP CODE
        LDA     INSPNT
        JSR     PRBYTS
        LDYIM $00
        LDXIM $0F       15 COLUMN OBJECT FIELD

OBJ     LDAIY INSPNT PRINT OP CODE AND OPERAND
        JSR     PRBYTS AS A FUNCTION OF LENGTH
        DEX
        DEX
        DEX             MINUS 1 BYTE AND 1 SPACE
        INY
        CPY     LENGTH OBJECT FINISHED?
        BNE     OBJ
        JSR     XSPACE FILLUP WITH SPACES
        RTS

*** OBJMNE ***

OBJMNE  JSR     OBJECT  PRINT A FORMATTED OBJECT CODE

PRMNES  LDXIM $03       AND THE CORRESPONDING MNEMONICS

MNEMON  LDYIM $05
```

```
        LDAIM $00     5 SHIFTS PER CHARACTER, RESET A

MNEA    ASL   MR      ENCODE MNEMONIC INTO A
        ROL   ML
        ROLA
        DEY
        BNE   MNEA
        ORAIM $40     RESTORE ASCII CODE
        JSR   PRCHA   PRINT MNEMONIC CHARACTER
        DEX
        BNE   MNEMON  3 MNEMONIC CHAR. PRINTED
        JSR   PRSP
        RTS

*** PRINT ***

PRINT   PLA           PULL RETURN ADDRESS FROM STACK
        STA   MEPNT   AND SAVE IT
        PLA
        STA   MEPNT   +01

PRTA    INC   MEPNT
        BNE   PRTB
        INC   MEPNT   +01

PRTB    LDYIM $00     FETCH CHARACTER AND
        LDAIY MEPNT   PRINT IT
        CMPIM $03     EOT?
        BEQ   PRTC
        JSR   PRCHA
        JMP   PRTA

PRTC    LDA   MEPNT   +01 PUSH RETURN ADDRESS ON
        PHA           STACK AND RETURN
        LDA   MEPNT
        PHA
        RTS

*** ADJ ***

ADJ     LDX   INSPNT ADJUST FOR REL. BRANCH
        LDY   INSPNT +01
        INX
        BNE   ADJA
        INY

ADJA    STX   ADDR
        STY   ADDR   +01 ACCU IS NOT CHANGED!
        RTS


**********
COMPRESSED MNEMONICS
**********
```

```
MALTAB =    $7C    ORA    ML PART
       =    $0B    AND    COLUMNS 1,5,9,D
       =    $2B    EOR
       =    $09    ADC
       =    $9D    STA
       =    $61    LDA
       =    $1B    CMP
       =    $98    SBC

MARTAB =    $82    ORA    MR PART
       =    $38    AND    COLUMNS 1,5,9,D
       =    $E4    EOR
       =    $06    ADC
       =    $02    STA
       =    $02    LDA
       =    $60    CMP
       =    $86    SBC

MBLTAB =    $0C    ASL    ML PART
       =    $93    ROL    COLUMNS 6,E,2(LDX# ONLY)
       =    $64    LSR
       =    $93    ROR
       =    $9D    STX
       =    $61    LDX    (INCLUDING LDXIM)
       =    $21    DEC
       =    $4B    INC

MBRTAB =    $D8    ASL    MR PART
       =    $D8    ROL    COLUMNS 6,E,2(LDX# ONLY)
       =    $E4    LSR
       =    $E4    ROR
       =    $30    STX
       =    $30    LDX    (INCLUDING LDXIM)
       =    $46    DEC
       =    $86    INC

MCLTAB =    $14    BRK    ML PART
       =    $14    BPL    COLUMN Ø
       =    $54    JSR    AND INVALID
       =    $13    BMI    OPCODE
       =    $95    RTI
       =    $15    BVC
       =    $95    RTS
       =    $15    BVS
       =    $00    ØØØ    NO VALID OPCODE
       =    $10    BCC
       =    $61    LDY
       =    $10    BCS
       =    $1C    CPY
       =    $13    BNE
       =    $1C    CPX
       =    $11    BEQ

MCRTAB =    $96    BRK    MR PART
       =    $18    BPL    COLUMN Ø
       =    $E4    JSR    AND INVALID
                          OPCODE
```

```
        =       $52     BMI
        =       $12     RTI
        =       $86     BVC
        =       $26     RTS
        =       $A6     BVS
        =       $00     ???   NO VALID OPCODE
        =       $C6     BCC
        =       $32     LDY
        =       $E6     BCS
        =       $32     CPY
        =       $8A     BNE
        =       $30     CPX
        =       $62     BEQ

MDLTAB  =       $FF     NOT USED
        =       $12     BIT     ML PART
        =       $53     JMP     COLUMN 4,C
        =       $53     JMP
        =       $9D     STY
        =       $61     LDY
        =       $1C     CPY
        =       $1C     CPX

MDRTAB  =       $FE     NOT USED
        =       $68     BIT     MR PART
        =       $60     JMP     COLUMN 4,C
        =       $60     JMP
        =       $32     STY
        =       $32     LDY
        =       $32     CPY
        =       $30     CPX

MELTAB  =       $0C     ASL             ML PART
        =       $FF     NOT USED COLUMN A
        =       $93     ROL
        =       $FF     NOT USED
        =       $64     LSR
        =       $FF     NOT USED
        =       $93     ROR
        =       $FF     NOT USED
        =       $A6     TXA
        =       $A6     TXS
        =       $A0     TAX
        =       $A4     TSX
        =       $21     DEX
        =       $FF     NOT USED
        =       $73     NOP
        =       $FF     NOT USED

MERTAB  =       $D8     ASL             MR PART
        =       $FE     NOT USED        COLUMN A
        =       $D8     ROL
        =       $FE     NOT USED
        =       $E4     LSR
        =       $FE     NOT USED
        =       $E4     ROR
```

```
        =       $FE     NOT  USED
        =       $02     TXA
        =       $26     TXS
        =       $70     TAX
        =       $F0     TSX
        =       $70     DEX
        =       $FE     NOT  USED
        =       $E0     NOP
        =       $FE     NOT  USED

MFLTAB  =       $82     PHP     ML PART
        =       $1B     CLC     COLUMN 8
        =       $83     PLP
        =       $99     SEC
        =       $82     PHA
        =       $1B     CLI
        =       $83     PLA
        =       $99     SEI
        =       $21     DEY
        =       $A6     TYA
        =       $A0     TAY
        =       $1B     CLV
        =       $4B     INY
        =       $1B     CLD
        =       $4B     INX
        =       $99     SED

MFRTAB  =       $20     PHP     MR PART
        =       $06     CLC     COLUMN 8
        =       $20     PLP
        =       $46     SEC
        =       $02     PHA
        =       $12     CLI
        =       $02     PLA
        =       $52     SEI
        =       $72     DEY
        =       $42     TYA
        =       $72     TAY
        =       $2C     CLV
        =       $B2     INY
        =       $08     CLD
        =       $B0     INX
        =       $48     SED

NVALA   =       $02             COLUMN 2   NO
        =       $03             COLUMN 3   VALID
        =       $07             COLUMN 7   OPCODES
        =       $0B             COLUMN B
        =       $0F             COLUMN F

NVALB   =       $80             ALL "BLANKS"
        =       $04             (INVALID OPCODES)
        =       $14             IN COLUMNS
        =       $34             Φ,4,9,A,C AND E
        =       $44
        =       $54
```

```
        =       $64
        =       $74
        =       $D4
        =       $F4
        =       $89
        =       $1A
        =       $3A
        =       $5A
        =       $7A
        =       $DA
        =       $FA
        =       $0C
        =       $1C
        =       $3C
        =       $5C
        =       $7C
        =       $9C
        =       $DC
        =       $FC
        =       $9E


************************************
JUMP TABLE TO EXTERNAL SUBROUTINES
************************************


CRLF    JMP     $11E8
PRCHA   JMP     $1334   OUTPUT SUBROUTINE
RECCHA  JMP     $12AE   INPUT SUBROUTINE
PRBYT   JMP     $128F   OUTPUT THE BYTE IN A
PRSP    JMP     $11F3   OUTPUT A SPACE
INPAR   JMP     $1387   INPUT 2 PARAMETERS
RESIN   JMP     $1268   RESET BUFFERS
INCPNT  JMP     $1213   INCREMENT A POINTER
PRNIBL  JMP     $129B   OUTPUT A NIBBLE
USR     JMP     $105F   USER EXIT

******************
FORMAT SUBROUTINES
******************


MAFORM  LDA     LNIB
        LSRA            DIVIDE INDEX BY 2
        TAX
        LDAX    MALTAB  FETCH COMPRESSED MNEMONICS
        STA     ML
        LDAX    MARTAB
        STA     MR

MNEMOC  JSR     OBJMNE  OUTPUT OBJECT AND MNEMONIC
        RTS

MBFORM  LDA     LNIB    DIVIDE INDEX BY 2
        LSRA
```

```
        TAX
        LDAX  MBLTAB
        STA   ML
        LDAX  MBRTAB FETCH COMPRESSED MNEMONIC
        STA   MR
        JMP   MNEMOC

MCFORM  LDX   LNIB

MCFA    LDAX  MCLTAB FETCH COMPR. MNEMONIC
        STA   ML
        LDAX  MCRTAB
        STA   MR
        JMP   MNEMOC

MDFORM  LDA   LNIB
        LSRA          DIVIDE INDEX BY 2
        TAX
        LDAX  MDLTAB FETCH COMPRESSED MNEMONIC
        STA   ML
        LDAX  MDRTAB
        STA   MR
        JMP   MNEMOC

MEFORM  LDX   LNIB
        LDAX  MELTAB FETCH COMPRESSED MNEMONIC
        STA   ML
        LDAX  MERTAB
        STA   MR
        JMP   MNEMOC

MFFORM  LDX   LNIB
        LDAX  MFLTAB FETCH COMPRESSED MNEMONIC
        STA   ML
        LDAX  MFRTAB
        STA   MR
        JMP   MNEMOC



*****************************
MONITOR OF THE DISASSEMBLER
*****************************


L: DISASSEMBLE FROM/TO
P: DISASSEMBLE 16 LINES (WINDOW)
 : SPACE BAR = DISASSEMBLE 1 LINE
D: START DISASSEMBLER
H: PRINT A HEXDUMP
A: PRINT AN ASCII DUMP
R: RELEASE THIS MONITOR


DISMON LDAIM DISMA  LOAD BRK VECTOR
       LDXIM DISMA  /256
```

```
        STA   BRKT
        STX   BRKT   +01
        JSR   PRINT
        =     $0D    CRLF
        =     $0A
        =     'V
        =     'A
        =     'L
        =     'I
        =     'D
        =     '
        =     'C
        =     'O
        =     'M
        =     'M
        =     'A
        =     'N
        =     'D
        =     'S
        =     ':
        =     '
        =     'A
        =     '
        =     'D
        =     '
        =     'H
        =     '
        =     'L
        =     '
        =     'P
        =     '
        =     'R
        =     '
        =     'S
        =     'P
        =     $0D
        =     $0A
        =     $03

DISMA   JSR   CRLF

DISMB   JSR   RECCHA WAIT FOR A KEY STROKE
        CMPIM 'D     D-COMMAND?
        BNE   DISMD
        JSR   PRINT
        =     $0D
        =     $0A
        =     'D
        =     'I
        =     'S
        =     'A
        =     'S
        =     'S
        =     'E
        =     'M
        =     'B
```

```
        =       'L
        =       'E
        =       ':
        =       '
        =       $03
        JSR     RESIN   RESET INPUT BUFFER
        JSR     INPAR   GET PARAMETERS
        BMI     DISMA   VALID CHARACTER?
        JSR     CHCK    1ST PAR < 2ND PAR?
        BCC     DISMA

DISMC   LDA     PARA
        LDX     PARA    +01
        STA     INSPNT  SET UP INSTRUCTION POINTER
        STX     INSPNT +01
        JSR     PRINT
        =       $0D
        =       $0A
        =       'L
        =       '
        =       'P
        =       '
        =       'S
        =       'P
        =       '
        =       '?
        =       $03
        JMP     DISMA

DISMD   CMPIM   'P      P COMMAND?
        BNE     DISMF
        LDAIM   $0F     SET LINECOUNTER AND PAGE MODE
        STA     PMODE
        STA     CNT

DISME   SEC
        LDA     PARB    INSPNT < PARB?
        SBC     INSPNT
        LDA     PARB    +01
        SBC     INSPNT +01
        BCC     DISMB
        JSR     DISASM DISASSEMBLE THE CURR. INSTR.
        LDA     PMODE   CHECK PAGE MODE
        BEQ     DISME
        DEC     CNT
        BNE     DISME   15 LINES?
        BEQ     DISMB

DISMF   CMPIM   'L      L COMMAND?
        BNE     DISMG
        LDAIM   $00
        STA     PMODE   RESET PAGE MODE
        BEQ     DISME

DISMG   CMPIM   '       SPACE BAR?
        BNE     DISMH
```

```
        LDAIM $01
        STA   PMODE  DISASSEMBLE 1 INSTRUCTION
        STA   CNT
        BNE   DISME

DISMH   CMPIM 'H     H COMMAND?
        BNE   DISMQV
        LDAIM $01
        STA   HEXFLG SET HEXFLAG
        JSR   PRINT
        =     $0D
        =     $0A
        =     'H
        =     'E
        =     'X
        =     '
        =     'D
        =     'U
        =     'M
        =     'P
        =     ':
        =     '
        =     $03

DISMR   JSR   RESIN
        JSR   INPAR
        BPL   DISMK  VALID INPUT?

DISMI   JMP   DISMA

DISMQV  JMP   DISMQ


DISMK   JSR   CHCK
        BCC   DISMI
        JSR   CRLF
        JSR   CRLF
        LDXIM $06
        JSR   XSPACE
        LDYIM $00

DISM    TYA          PRINT HEADER
        JSR   PRNIBL OUTPUT A NIBBLE
        LDXIM $02
        JSR   XSPACE
        INY
        CPYIM $10    16 COLUMNS?
        BNE   DISML
        LDA   PARA
        STA   POINT  SET DUMP POINTER
        LDA   PARA   +01
        STA   POINT  +01
        JSR   CRLF

DISMP   JSR   CRLF
        LDXIM $10
```

```
        STX   CNT
        LDA   POINT  +01
        JSR   PRBYT
        LDA   POINT
        JSR   PRBYT  OUTPUT CURRENT ADDRESS
        JSR   PRINT
        =     ':
        =     '
        =     $03

DISMN   LDA   PARB
        SEC
        SBC   POINT
        LDA   PARB   +01
        SBC   POINT  +01
        BCS   DISMO  HEX DUMP FINISHED?
        JMP   DISMA

DISMO   LDYIM $00
        LDAIY POINT
        LDX   HEXFLG
        BEQ   DISMT  IS HEX FLAG SET?
        JSR   PRBYT

DISMU   JSR   PRSP
        JSR   INCPNT
        DEC   CNT
        BNE   DISMN  16 COLUMNS PRINTED?
        BEQ   DISMP

DISMT   CMPIM $20    ASCII FILTER
        BCC   DISMV
        CMPIM $7F
        BCS   DISMV
        JSR   PRCHA  OUTPUT ASCII
        LDXIM $01

DISMW   JSR   XSPACE
        JMP   DISMU

DISMV   LDXIM $02
        BNE   DISMW

DISMQ   CMPIM 'A     A COMMAND?
        BNE   RELEAS
        LDAIM $00
        STA   HEXFLG
        JSR   PRINT
        =     $0D
        =     $0A
        =     'A
        =     'S
        =     'C
        =     'I
        =     'I
        =     '
```

```
=       'D
=       'U
=       'M
=       'P
=       ':
=       '
=       $03
JMP     DISMR

DISMS   JMP     DISMA

CHCK    LDA     PARB    PARA < PARB?
        SEC
        SBC     PARA
        LDA     PARB    +01
        SBC     PARA    +01
        RTS

RELEAS  CMPIM   'R      R COMMAND?
        BNE     DISMS
        JMP     USR     USER SELECTABLE ADDR.

**********
END OF PROGRAM
**********
```

**********ONE**********

```
        ORG $FDDA

        EPROM PROGRAMMING UTILITIES

        WRITTEN BY G.H.NACHBAR

        DATE:26 JANUARY 1982

POINTERS AND TEMPS IN PAGE ZERO
CMPMOD  *   $0028   COMPARE FLAG
SORSAL  *   $00E2   FIRST SOURCE ADDRESS LOW
SORSAH  *   $00E3   FIRST SOURCE ADDRESS HIGH
SOREAL  *   $00E4   LAST SOURCE ADDRESS LOW
SOREAH  *   $00E5   LAST SOURCE ADDRESS HIGH
CURADL  *   $00E6   SOURCE POINTER LOW
CURADH  *   $00E7   SOURCE POINTER HIGH
DESSAL  *   $00E8   FIRST DESTINATION ADDRESS LOW
DESSAH  *   $00E9   FIRST DESTINATION ADDRESS HIGH
DIFL    *   $00EA   (DESSA MINUS SORSA) LOW
DIFH    *   $00EB   (DESSA MINUS SORSA) HIGH
BYTES   *   $00F6   INSTRUCTION LENGTH BUFFER
POINTL  *   $00FA   DESTINATION POINTER LOW
POINTH  *   $00FB   DESTINATION POINTER HIGH

POINTERS AND TEMPS IN PAGE 1A
PARAL   *   $1A63   FIRST ADDRESS LOW
PARAH   *   $1A64   FIRST ADDRESS HIGH
PARBL   *   $1A65   SECOND ADDRESS LOW
PARBH   *   $1A66   SECOND ADDRESS HIGH
NMIL    *   $1A7A   NMI VECTOR LOW
NMIH    *   $1A7B   NMI VECTOR HIGH

ADRESSES IN STANDARD EPROM
BEGIN   *   $1ED3   INITIALIZE SOURCE POINTER
OPLEN   *   $1E5C   GET INSTRUCTION LENGTH

ADDRESSES IN PM/PME EPROM
IPB     *   $13A2   ENTER ONE ADDRESS
INCPNT  *   $1213   INCREMENT POINT
PRBUFS  *   $11F8   PRINT DATA SPECIFIED BY POINT
```

**********TWO**********

ADDRESSES ELSEWHERE IN DISASSEM/EPRUTL EPROM

USR	*	$FBEA	JUMP TO PM
INPAR	*	$FBDE	ENTER TWO ADDRESSES
RESIN	*	$FBE1	RESET INPUT BUFFERS
PRINT	*	$FAF2	PRINT DATA STRING FOLLOWING JSR PRINT
CRLF	*	$FBCF	FRESH LINE
RECCHA	*	$FBD5	WAIT FOR AN ASCII CHARACTER
CHCK	*	$FDC5	COMPARE THE TWO INPAR ADDRESSES

****

KEY ROUTINES

****

*1 P KEY

ENTER SORSA,SOREA AND DESSA

*2 M KEY

DATA WITH AN ADRESS WITHIN THE SOURCE DATABLOCK IS COPIED INTO LOCATIONS FROM DESSA ONWARDS (SEE NOTE)

*3 B KEY

BACK TO PM

*4 V KEY

COMPARE DESTINATION DATA WITH SOURCE DATA. IF NOT EQUAL: PRINT DESTINATION DATA WITH ADDRESS (SEE NOTE)

*5 F KEY

EPROM CHECK:IF DESTINATION DATA IS NOT $FF,PRINT IT WITH ADDRESS (SEE NOTE)

*6 R KEY

ALL OPERANDS OF THREE BYTE INSTRUCTIONS WITH SORSA<=OPERAND=< SOREA ARE CHANGED ACCORDING TO THE NEW DESTINATION OF THE SOURCE DATA BLOCK (SEE NOTE)

*7 NMI/ST KEY (STANDARD KEYBOARD)

PRINT PARAMETERS

NOTE: THE DIFFERENCE BETWEEN CURAD AND SORSA ALWAYS EQUALS THE DIFFERENCE BETWEEN POINT AND DESSA

****

MAIN ROUTINE OF THE EPROM PROGRAMMING SOFTWARE

****

EPRUTL	LDAIM PRMTRS	SPECIFY NMI VECTOR
	LDYIM PRMTRS/256	

```
*THREE**********

 STA    NMIL
 STY    NMIH
 JSR    PRINT          INITIAL BLURP
 $OD
 $OA
 'E
 'P
 'R
 'O
 'M
 '
 'P
 'R
 'O
 'G
 'R
 'A
 'M
 'M
 'I
 'N
 'G
 '
 'U
 'T
 'I
 'L
 'I
 'T
 'I
 'E
 'S
 $OD
 $OA
 'V
 'A
 'L
 'I
 'D
 '
 'C
 'O
 'M
 'M
 'A
 'N
 'D
 'S
 ':
 '
 'P
 '
 'M
 '
 'B
 '
 'V
 '
 'F
 '
 'R
 $03
```

```
*FOUR*************
JSR     CRLF            START ON A FRESH LINE
JSR     RECCHA          WAIT FOR A DEPRESSED KEY
CMPIM   'P
BNE     EPRD            BRANCH IF NO P KEY
JSR     PRINT           SOURCE MESSAGE
$OD
$OA
'F
'I
'R
'S
'T
'
',
'L
'A
'S
'T
'
'S
'O
'U
'R
'C
'E
'
'A
'D
'D
'R
'E
'S
'S
':
'
'
$O3                     EOT
JSR     RESIN           RESET INPUT BUFFERS
JSR     INPAR           ENTER TWO ADDRESSES
BMI     EPRB            REPEAT IF NOT DONE PROPERLY
JSR     CHCK
BCC     EPRB            REPEAT IF LAST<FIRST ADDRESS
LDA     PARAL
LDY     PARAH
STAZ    SORSAL
STYZ    SORSAH          SORSA=FIRST ADDRESS ENTERED
LDA     PARBL
LDY     PARBH
STAZ    SOREAL
STYZ    SOREAH          SOREA=SECOND ADDR. ENTERED
JSR     PRINT           SOURCE MESSAGE
$OD
$OA
'F
'I
'R
'S
'T
'
'D
'E
```

```
*****FIVE**************
        'S
        'T
        'I
        'N
        'A
        'T
        'I
        'O
        'N
        '
        'A
        'D
        'D
        'R
        'E
        'S
        'S
        ':
        '
        '
        '
        $03                             EOT
        JSR     RESIN                   RESET INPUT BUFFERS
        JSR     IPB                     ENTER ONE ADDRESS
        BMI     EPRC                    REPEAT IF NOT PROPERLY DONE
        LDA     PARBL
        LDY     PARBH
        STAZ    DESSAL
        STYZ    DESSAH                  DESSA=ADDRESS ENTERED
        JMP     EPRA                    READY FOR NEW USER ACTION
EPRD    CMPIM   'M
        BNE     EPRE                    BRANCH IF NO M KEY
        JSR     BEGIN                   CURAD=SORSA
        JSR     FIRST                   POINT=DESSA
EPRO    LDYIM   $00
        LDAIY   CURADL                  FETCH SOURCE DATA
        STAIY   POINTL                  MOVE IT TO DESTINATION
        JSR     INCPNT                  ADJUST DESTINATION POINTER
        LDAIM   $01
        STAZ    BYTES
        JSR     NXT                     ADJUST SOURCE POINTER
        BCS     EPRO                    NEXT DATA,IF ANY
        JSR     PRINT                   FINAL MESSAGE
        $OD
        $OA
        'D
        'A
        'T
        'A
        '
        'M
        'O
        'V
        'E
        'D
        $03                             EOT
        JMP     EPRA                    READY FOR NEW USER ACTION
EPRE    CMPIM   'B
        BNE     EPRF                    BRANCH IF NO B KEY
```

```
**SIX***************
JMP    USR          BACK TO PM ("JUNIOR")
CMPIM  'V
BNE    EPRG         BRANCH IF NO V KEY
LDAIM  $00          GET CMPMOD
STAZ   CMPMOD       RESET CMPMOD
JSR    BEGIN        CURAD=SORSA
JSR    FIRST        POINT=DESSA
LDYIM  $00
LDAZ   CMPMOD
BNE    EPRL         BRANCH IF CMPMOD IS SET
LDAIY  CURADL       FETCH SOURCE DATA
CMPIY  POINTL       COMPARE IT WITH DEST. DATA
BEQ    EPRN         IF EQUAL:NEXT DATA COMPARE
JSR    PRBUFS       PRINT UNEQUAL DEST. DATA
JSR    INCPNT       ADJUST DESTINATION POINTER
LDAIM  $01
STAZ   BYTES
JSR    NXT          ADJUST SOURCE POINTER
BCS    EPRK         NEXT DATA,IF ANY
JSR    PRINT        FINAL MESSAGE
$0D
$0A
'D
'A
'T
'A
'
'C
'O
'M
'P
'A
'R
'E
'D
$03                 EOT
JMP    EPRA         READY FOR NEW USER ACTION
CMPIM  'F
BNE    EPRI         BRANCH IF NO F KEY
LDAIM  $01
STAZ   CMPMOD       SET CMPMOD
BNE    EPRH         GO TO COMPARE ROUTINE
LDAIY  POINTL       FETCH DESTINATION DATA
CMPIM  $FF          AND COMPARE IT WITH $FF
JMP    EPRM         BACK TO COMPARE ROUTINE
CMPIM  'R
BNE    EPRJ         BRANCH IF NO R KEY
JSR    BEGIN        CURAD=SORSA
JSR    FIRST        POINT=DESSA
JSR    DIFAD        DIF IS DESSA MINUS SORSA
JSR    OPLEN        GET INSTRUCTION LENGTH
CPYIM  $03
BNE    NXTINS       BRANCH IF NO 3 BYTE INSTRUCTION
DEY
DEY
LDAIY  CURADL       FETCH ADL
SEC
```

```
**********SEVEN**************

                SBCZ   SORSAL      ADL MINUS SORSAL
                INY
                LDAIY  CURADL      FETCH ADH
                SBCZ   SORSAH      ADH MINUS SORSAH MINUS C
                BCC    NXTINS      BRANCH IF OPERAND < SORSA
                DEY
                LDAZ   SOREAL      FETCH LAST SOURCE ADDRESS LOW
                SBCIY  CURADL      SOREAL MINUS ADL
                INY
                LDAZ   SOREAH      FETCH LAST SOURCE ADDRESS HIGH
                SBCIY  CURADL      SOREAH MINUS ADH MINUS C
                BCC    NXTINS      BRANCH IF OPERAND > SOREA
                CLC
                DEY
                LDAIY  CURADL      FETCH OLD ADL
                ADCZ   DIFL        AND ADD DIFL TO IT
                STAIY  CURADL      REPLACE ADL (SOURCE DATA BLOCK)
                STAIY  POINTL      REPLACE ADL (DEST. DATA BLOCK)
                INY
                LDAIY  CURADL      FETCH OLD ADH
                ADCZ   DIFH        ADD DIFH AND C TO IT
                STAIY  CURADL      REPLACE ADH (SOURCE DATA BLOCK)
                STAIY  POINTL      REPLACE ADH (DEST. DATA BLOCK)
NXTINS          LDYZ   BYTES
NXTIN           JSR    INCPNT      ADJUST DESTINATION POINTER
                DEY
                BNE    NXTIN
                JSR    NXT         ADJUST SOURCE POINTER
                BCS    TEST        NEXT INSTRUCTION,IF ANY
                JSR    PRINT       FINAL MESSAGE
                $OD
                $OA
                'R
                'E
                'L
                'O
                'C
                'A
                'T
                'E
                'D
                $03                EOT
          EPRJ  JMP    EPRA        READY FOR NEW USER ACTION

**********************************************************
SUBROUTINES OF THE EPROM PROGRAMMING SOFTWARE
**********************************************************

          FIRST LDAZ   DESSAL      INITIALIZE DESTINATION POINTER
                LDYZ   DESSAH
                STAZ   POINTL
                STYZ   POINTH      DESTINATION POINTER=DESSA
                RTS
```

**********EIGHT**********

SUBROUTINE DIFAD

THE 16 BIT NUMBER (DIFH,DIFL) EQUALS THE DIFFERENCE BETWEEN (DESSAH,DESSAL) AND (SORSAH,SORSAL). IT IS USED IN THE R KEY ROUTINE

```
DIFAD   SEC
        LDAZ   DESSAL
        SBCZ   SORSAL
        STAZ   DIFL
        LDAZ   DESSAH
        SBCZ   SORSAH
        STAZ   DIFH          DIF=DESSA MINUS SORSA
        RTS
```

SUBROUTINE NXT

THE SOURCE POINTER CURAD IS INCREASED BY THE CONTENTS OF BYTES. THE CONTENTS OF BYTES IS EITHER AN INSTRUCTION LENGTH (R KEY ROUTINE) OR $01 (M,V AND F KEY ROUTINE).
SUBSEQUENTLY THE CARRY FLAG WILL BE SET OR RESET,DEPENDING ON THE NEW POSITION (CONTENTS) OF CURAD.
AFTER RTS THE CARRY FLAG WILL BE
*SET (C=1;$\overline{B}$=0) IF NEW CURAD IS SMALLER THAN OR EQUALS SOREA;
*RESET (C=0;$\overline{B}$=1) IF NEW CURAD IS GREATER THAN SOREA,
OR IF THE 65K MEMORY BORDER ($FFFF) IS CROSSED AT THE COMPUTATION OF THE NEW CURAD. THE LATTER OCCURS IF THE OLD CURADH IS $FF AND IF THE OLD CURADL IS $FF (M,V AND F KEY ROUTINE) OR $FE OR $FD (R KEY ROUTINE)

```
NXT     CLC
        LDAZ   CURADL
        ADCZ   BYTES
        STAZ   CURADL
        LDAZ   CURADH
        ADCIM  $OO
        STAZ   CURADH        CURAD INCREASED BY (BYTES)
        BCS    NXTB          BRANCH IF $FFFF IS CROSSED
        SEC
        LDAZ   SOREAL        CARRY
        SBCZ   CURADL        DEPENDS
        LDAZ   SOREAH        ON
        SBCZ   CURADH        SOREA MINUS CURAD
NXTA    RTS                  OR
NXTB    CLC                  ON CROSSING $FFFF,
        BCC    NXTA          THE MEMORY BOUNDARY
```

NOTE: WHY ALL THE FUSS ABOUT CROSSING $FFFF? WELL,IF WE SKIP THE BCS NXTB,THE FOLLOWING CURAD TEST WILL BE INVALID,BECAUSE,IF SOREA IS $FFFF,THE NEW CURAD WILL BE SMALLER THAN SOREA. IT IS LIKE MAKING A TRIP AROUND THE WORLD WHICH WILL LAST FOREVER,BECAUSE WE DIDN'T NOTICE THE FACT THAT WE HAVE BEEN PASSING OUR DEPARTING POINT!

**********NINE**********

NMI/ST KEY ROUTINE PRMTRS

PRINT "XXXX< =AD=< YYYY TO > =ZZZZ"
XXXX: FIRST SOURCE ADDRESS
YYYY: LAST SOURCE ADDRESS
ZZZZ: FIRST DESTINATION ADDRESS

A GRAFICAL SUMMARY OF THE MOST RECENT ADDRESS PARAMETERS

```
PRMTRS  PLA
        PLA
        PLA                     RESTORE STACK POINTER
        JSR     CRLF            START ON A FRESH LINE
        LDAZ    SORSAH
        JSR     PRBYT           PRINT SORSAH
        LDAZ    SORSAL
        JSR     PRBYT           PRINT SORSAL
        JSR     PRINT           SOURCE MESSAGE
        '<
        '=
        'A
        'D
        '=
        '<
        $03                     EOT(END OF ASCII STRING)
        LDAZ    SOREAH
        JSR     PRBYT           PRINT SOREAH
        LDAZ    SOREAL
        JSR     PRBYT           PRINT SOREAL
        JSR     PRINT           DESTINATION MESSAGE
        '
        'T
        'O
        '
        '>
        '=
        $03                     EOT(END OF ASCII STRING)
        LDAZ    DESSAH
        JSR     PRBYT           PRINT DESSAH
        LDAZ    DESSAL
        JSR     PRBYT           PRINT DESSAL
        JMP     EPRA            READY FOR NEW USER ACTION
```

****

SYSTEM VECTORS

****

```
        $2F     ADL OF JMI (NMI JUMP VECTOR)
        $1F     ADH OF JMI (NMI JUMP VECTOR)
        $1D     ADL OF RESET (ST. EPROM)
        $1C     ADH OF RESET (ST. EPROM)
        $32     ADL OF JMI  (IRQ JUMP VECTOR)
        $1F     ADH OF JMI  (IRQ JUMP VECTOR)
```